# Kattis Team Guide

version 1.1

## Table of Contents

# Getting started with KATTIS

Kattis, the KTH Automated Teaching Tool, is a server package to correct ACM style problems. During this competition you will be submitting your code to Kattis for judgment. This document provides an introduction to Kattis and covers the most common questions about Kattis.

# Connecting to KATTIS

In order to access Kattis, you open a web-browser and go to the Kattis homepage[1]. At this point you can log in, using the login and password supplied to you.

# Solving a simple problem

One problem typically found in Kattis is the problem “hello”. Hello is a version of the classic “Hello World!” Program. It is very simple: print the line “Hello World!”. Some sample solutions are:

```
/* C: */
#include <stdio.h>
main() {
  printf("Hello World!\n");
  return 0;
}
```

```
/* C++: */
#include <iostream>
main() {
  std::cout << "Hello World!" <<std::endl;
  return 0;
}
```

```
/* Java: */
public class hello{
  public static void main(String[] args){
    System.out.println("Hello World!");
    return;
  }
}
```

Type in the program in your favorite text editor and save it. You might also want to compile and test run the program locally. There are two ways to submit code to Kattis. You can do it over the homepage, or you can use the submit script.

# Using the submit script

The submit script (available on your machine as submit) allows you to easily submit your program from the command line. If you name the file after the problem (hello.java) it will guess the problem as hello and the main class as hello. If the guesses are wrong you can correct them using flags to the script. The submit script also allows you to submit multiple files if you need to.

[1] The organizers will provide a URL to Kattis.

A typical session using the submit script looks like this:

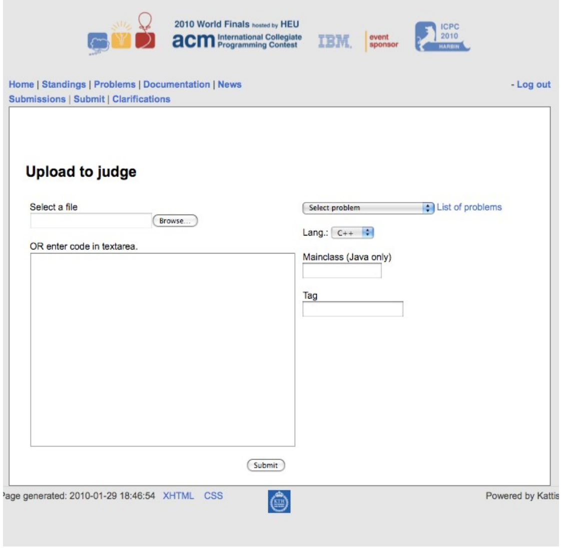
```
Usage: submit.py [options]

Options:
  -h, --help            show this help message and exit
  -p PROBLEM, --problem=PROBLEM
                        Submit to problem PROBLEM. Overrides default guess
                        (first part of first filename)
  -m CLASS, --mainclass=CLASS
                        Sets mainclass to CLASS. Overrides default guess
                        (first part of first filename)
  -l LANGUAGE, --language=LANGUAGE
                        Sets language to LANGUAGE. Overrides default guess
                        (based on suffix of first filename)
  -t TAG, --tag=TAG     Sets submission TAG (arbitary string associated with
                        the submission).
  -f, --force           Force, no confirmation prompt before submission
  -d, --debug           Print debug info while running

$ submit hello.java
Problem: hello
Language: Java
Files: hello.java
Mainclass: hello
Submit (y/N)?
y
Submission recieved. Submission ID: 1445
```

# Web submissions

Another way to submit your program to Kattis is to log into the web interface and choose submit. Here you have a web form where you can submit the file.


2010 World Finals hosted by HEU
acm International Collegiate Programming Contest
IBM
event sponsor
ICPC 2010 HARBIN
Home | Standings | Problems | Documentation | News
Submissions | Submit | Clarifications
- Log out
Browse...
Select problem
List of problems
Submit
XHTML
CSS

Upload to judge

Select a file

OR enter code in textarea.

Lang.: C++

Mainclass (Java only)

Tag

Page generated: 2010-01-29 18:46:54

Powered by Kattis

# How does KATTIS handle a submitted program?

First, Kattis will compile your program. If the compiler fails to compile your program, or does not complete in a reasonable amount of time, she will judge it as Compile Error. Otherwise, she will execute the compiled binary on the first input file (there may be several). If the execution takes too long it will be judged as Time Limit Exceeded, if it crashes, returns a non-zero exit code, or uses too much memory it will be judged as a Run Time Error . If the execution terminates correctly (*return code 0*), she will inspect the output produced to verify that it is correct. If it is incorrect, she will judge the submission as Wrong Answer.

Note that your submission's output is only inspected if it successfully terminates in time, thus a Time Limit Exceeded or Run Time Error does not mean that output was correct up until the point where the program crashed or time ran out.

If there are multiple test files, she will continue with the same procedure for the next test file, as long as no error has been found. As soon as an error is detected she will stop and report that error. Each test file is run with a new invocation of your program, so your program does not need to be adapted in any way to handle multiple test files. Each test file will follow the input specification for the problem.

If your program passes all tests files successfully, it will be judged as Accepted.

## Tracking the status of your submissions

You can track the status of your submission by logging in to the web interface and choosing Submissions. Here you will see a list of all submissions you have made, in reverse chronological order. As the submission proceeds through the judgment process the homepage will reflect this. The states a submission will pass through while it is being judged are:

New → Waiting for Compile → Compiling → Waiting for Run → Running → <Result>

## Clarifications

During the contest you can submit requests for clarifications to the judges. To do so, click on Clarifications. The clarifications page has three sections:

1. submitted clarification requests from your teams that have not yet been answered
2. a form for submitting a clarification request
3. clarification requests with answers from the judges

When you submit a clarification request, please select a subject (either one of the problems or “general”) and write your request in English.
The third section contains answers to your requests and sometimes answers to other teams' requests, in case the judges choose to reveal the question and the answer to all teams.
Notifications that there are new clarification replies are displayed on all Kattis web pages, but not on other pages (e.g., API documentation and scoreboards).

# Judge replies to submissions

When Kattis has judged your submission, you will get a reply telling you the status of your submission. You can view this reply on the homepage of Kattis. The following replies are possible:

### Submission Error

Submission Error means that Kattis did not understand the submission. This can be due to username/password error, not finding any files in the submission, seeing an invalid/missing language or problem tag, etc. At the world finals your computer will be configured to supply the correct username and password for the submission script.
A submission error *does not* give you any penalty time.

### Compile Error

Compile Error means that we failed to compile your source code. Extra information (with compiler

output) can be found on the page for the submission which can help you debug the error.
Information about what compilers and flags are used can be found on our technical info page.
Giving an incorrect, or not public, main class for a java program will cause a compile error!
Compile times over 1 minute will cause a compile error.
A compile error *does not* give you any penalty time.

## Judge Error

Judge Error means that you've found a bug (or at least misconfiguration) in the judge. This should never happen during the competition.
A judge error *does not* give you any penalty time.

## Wrong Answer

Wrong Answer means that your program finished within the time limit, but that the answer produced was incorrect. Make sure you follow the output specification to ensure that you don't get this error due to simple white-space errors.
Wrong answer gives penalty time

## Run Time Error

Run Time Error means that your program has crashed during execution with our secret test input (e.g. process exited with exit status ≠ 0 or the process was signaled). Make sure you return 0 from your program to not get an RTE. In java uncaught exceptions or termination with a non-zero exit status will be judged as Run Time Errors.[2]
Run time error gives penalty time.

## Time Limit Exceeded

Time Limit Exceeded means that your program ran for too long. When the time limit is exceeded, the program is terminated. The output produced is not inspected in this case, so getting Time Limit Exceeded does not mean that you produced correct output, only that your program did not exit in time.
Time limit exceeded gives penalty time.

## Illegal Function

Illegal Function means that your program tried to do something which it is not allowed to do when running on Kattis. This includes manipulation of file descriptors, opening files, running fork(), or exec(), creating threads, sending signals, installing signal handlers or basically anything except for solving the problem. If you think this is incorrect, please send in a clarification request.
An illegal function *does not* give you any penalty time. However, all illegal functions are examined and if it is determined to be a try to exploit the system the team may be disqualified from the competition.

## Accepted

Accepted means that your program has terminated successfully within the time limit and produced correct output. Congratulations!

[2] In C and C++, ensure that you return 0 from main(). In Java you don't need to do anything to ensure that the exit status is 0.

# Technical information

This is a technical description of how Kattis works, read this carefully.

# Input/Output

Your program should read its input from standard input (file descriptor 0, STDIN_FILENO/stdin in C, cin in C++, System.in in Java) and produce output on standard output (file descriptor 1, STDOUT_FILENO/stdout in C, cout in C++, System.out in Java). Anything written on standard error (file descriptor 2, STDERR_FILENO/stderr in C, cerr in C++, System.err in Java) will be ignored, this can be used for debugging your program during development (i.e., you do not have to remove debug output before submitting if you use standard error for debug output). Of course, writing to standard error will take some run-time. You can mix reading and writing freely (all input in an input file will be available when your program starts). Our input will always conform to the input specification. If, for example, the problem statement specifies that an integer is positive, you do not need to check for or handle non-positive integers (or non-integers). Always read the specification carefully.

## Example

When executing the program hello with the input file in.txt and output file out.txt the execution Kattis does will be identical to **$ hello < in.txt > out.txt**. After execution has finished Kattis will examine **out.txt** and compare it to the judge solution.

# Compilers

The following compilers and compiler flags are used when compiling your submissions.

| Language | Compiler | Version | Flags |
|---|---|---|---|
| C | gcc | 4.4.3 | -g -O2 -std=gnu99 -static <files> -lm |
| C++ | g++ | 4.4.3 | -g -O2 -static <files> |
| Java | OpenJDK | 1.6.0_20 | -O -sourcepath . -d . <files> |

When compiling, files with the following suffixes will be listed on the command line:

| Language | Suffixes |
|---|---|
| C | .c |
| C++ | .cc, .cpp, .cxx, .c++, .C |
| Java | .java |

Any other files will be ignored and most likely give you a compile error.
So if you submitted `foo.cc`, `foo.h`, and `main.cpp` the compiler will be run as `g++ -O2 -static foo.cc main.cpp`. Note that foo.h is not added to the commandline.[3] For all languages, a maximum compile time of 1 minute is enforced. If compilation takes longer than that, it will cause a compile error.
For java, the JVM used is the one included in OpenJDK 1.6.0_20. We increase the stack size limit to 8 megabytes (i.e. add the -Xss8m flag), increase the heap size (i.e. add the -Xmx2048Mb flag), and add the `-client` flag. So, if you submit a java program consisting of the file `hello.java` it

[3] Submitting multiple files is only possible if you use the submit script. Using the web interface, you can only submit a single file.

will be compiled as `″javac -O -sourcepath . -d . hello.java″,` and executed with `″java -client -Xss8m -Xmx2048Mb hello‶`
For C and C++, the heap size is set to match the Java heap size.

## System libraries

You are allowed to use all standard libraries included with the language you are working in. This includes the STL (Standard Template Library) for C++ and classes such as java.lang.BigInteger in Java. The STL version available is the version which shipped with the installed GCC version.

# Some FAQ

### What hardware is Kattis using?

Kattis is currently running on Hardware identical to your contest machines.

### What operating system is KATTIS using?

Kattis is currently running on the Ubuntu 10.04 distribution, fully patched.

# Important technical notices that can cost you the competition

### Must we be careful about our exit status?

Yes, if you exit with anything except for exit status 0, your submission will be judged as a Run Time Error. So make sure you return 0 at the end of your program.

### Must our Java main-class be public?

Yes, your main-class (as well as main method) must be *public*. This is due to the class loader used. Default is protected, so you will have to explicitly declare it public. Otherwise your submission will be judged as a Compile Error.

```
public class hello{
    public static void main(String[] args){
      System.out.println("Hello World!");
      return;
    }
}
```

```
class hello{
    public static void main(String[] args){
      System.out.println("Hello World!");
      return;
    }
}
```

Here the program on the left will compile and run correctly on Kattis, while the one on the right will give a compile error.

### Can I name my files what I want?

You have to use valid file endings for the language selected (see notes under compilers). File names may only consist of alphanumerical ascii characters, plus periods (“.”), and underscores (“_”), with the additional rule that the first and the last characters of the file name must both be alphanumerical. We strongly recommend naming the files after the name of the problems to simplify submissions using the submit script.